Long Read based Human Genomic Structural Variation Detection with cuteSV

Tao Jiang[1], Bo Liu[1], Yue Jiang[2], Junyi Li[1], Yan Gao[1], Zhe Cui[1], Yadong Liu[1] and Yadong Wang[1, *]

[1] Center for Bioinformatics, School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China

[2] Nebula Genomics, Harbin, Heilongjiang 150030, China

[*] Correspondence: ydwang@hit.edu.cn

Abstract

Long-read sequencing enables the comprehensive discovery of structural variations (SVs). However, it is still non-trivial to achieve high sensitivity and performance simultaneously due to the complex SV signatures implied by the noisy long reads. Herein, we propose cuteSV, a sensitive, fast and scalable long read-based SV detection approach. cuteSV uses tailored methods to comprehensively collect the signatures of various types of SVs, and a clustering-and-refinement method to analyze the signatures to implement a stepwise, highly sensitive SV detection. Benchmark on real PacBio and ONT datasets demonstrate that cuteSV has better yields as well as outstanding speed and scalability. cuteSV is available at https://github.com/tjiangHIT/cuteSV.

Keywords

Structural variants detection, Long-read sequencing, High sensitivity, Scalability

Background

Structural variations (SVs) represent genomic rearrangements, such as deletions, insertions, inversions, duplications, and translocations whose sizes are larger than 50 bp [1]. As the largest divergences across human genomes [2], SVs are close related to human diseases (e.g. inherited diseases [3-5] and cancers [6]), evolution (e.g. gene losses and transposon activity [7, 8]), gene regulations (e.g. rearrangements of transcription factors [9]) as well as many other phenotypes (e.g. mating and intrinsic reproductive isolation [10, 11]).

Many efforts have been made to develop short read-based SV calling approaches [12, 13]. State-of-the-art short read-based tools use various kinds of signatures, such as read-depths [14], discordant read-pairs [15], split read alignments [16], local assembly [17], or their combinations [18-20], and they have played important roles in many large-scale genomics studies, such as 1000 Genome Project [1]. However, the relatively poor read-length (typically a few hundreds of bps) limits these tools to implement highly sensitive SV detection [21] and there are still many false positives in their call sets [22].

With the rapid development of long read sequencing technologies, such as Pacific Bioscience (PacBio) [23] and Oxford Nanopore Technology (ONT) [24] platforms, read length is greatly increased, i.e., the average read length has been over 10 kbp. Such long-range spanning information provides the opportunity to comprehensively detect SVs at a high resolution [25]. However, the computational challenge still remains due to the higher sequencing error rates of long reads (typically 5-20%) [26]. That is, the aligners could produce less sensitive, chimeric and heterogeneous alignments for the reads around SV breakpoints, and the aligners could also have highly divergent behaviors for various types of SVs. Thus, the SV signatures implied by long read alignments are highly complicated and it is non-trivial to collect and analyze them to implement sensitive detection for various kinds of SVs.

Several long read alignment-based SV callers have been proposed in recent years, such as PB-Honey [27], SMRT-SV [28], Sniffles [29], PBSV (https://github.com/PacificBiosciences/pbsv) and SVIM [30]. Mainly, they use various methods to find SV evidences implied by the read alignments, such as the identification of local genomic regions with highly divergent alignments, the local assembly and re-alignment of clipped read parts, the clustering of SV-spanning signatures. Moreover, state-of-the-art long read aligners, such as BLASR [31], NGMLR [29], Minimap2 [32] and PBMM2 (https://github.com/PacificBiosciences/pbmm2), are usually employed to generate read alignments as the inputs to the SV callers.

Nevertheless, state-of-the-art long read-based SV callers still have some drawbacks: 1) overall, the sensitivity is still not satisfying, i.e., a high sequencing coverage is required and/or some SVs are still hard to be detected; 2) some methods like rMETL [33] and npInv [34] can only detect a subset or a particular class of SVs due to their specific design; 3) some tools like PBSV and SMRT-SV are still time-consuming and require many computational resources, which could be not very scalable and suited to many large-scale datasets; 4) some approaches like SMRT-SV and PB-Honey only support one type of sequencing data, e.g., only for PacBio reads, as they take the advantage of the characteristics of the data. These are still bottlenecking to the wide use of long read sequencing data in cutting-edge genomics studies and clinical practices.

Herein, we present cuteSV, a sensitive, fast and lightweight long read-based SV detection approach. This approach has several features: 1) cuteSV has better SVs detection yields than that of the state-of-the-art SV callers, moreover, it also has higher sensitivity on low coverage datasets, which is helpful to reduce the cost of sequencing. 2) cuteSV is a versatile SV caller which supports to process the datasets produced by mainstream platforms with various error rates and discover various types of SVs including

deletions, insertions, duplications, inversions and translocations. 3) With its tailored implementation, cuteSV also has faster or comparable speed to state-of-the-art approaches. 4) cuteSV has outstanding scalability, i.e., it enables to run with a low memory use and achieve a nearly linear speedup with the number of CPU threads, which is very suited to large-scale data analysis tasks. With these features, we believe that cuteSV has many potentials to cutting-edge genomics studies.

Results

Overview of cuteSV

Due to the high sequencing error and the complexity of SVs, the SV signals implied by the alignment of SV-spanning long reads are highly complicated. Under such circumstance, it requires SV detection approaches to have strong ability to well-handle various kinds of cases. Mainly, the following three technical issues should be fully considered.

1) Various types of SVs have different signatures, which can be represented as various combinations of large insertions/deletions as well as split alignments. Comprehensive SV signature extraction methods are needed to capture these signals. Especially, some large SV events could be divided as several smaller insertions/deletions in a local region, due to the scoring systems of read aligners, and they need to be well-handle to enhance the evidence of the real SV events.

2) Also due to the scoring systems of read aligners, the reads spanning the same SV could have heterogeneous breakpoints in the alignments. Robust read clustering methods are needed to handle the heterogeneous breakpoints to effectively cluster the reads spanning the same SVs and prevent multiple false positive clusters for one SV.

3) In some cases, there are multiple SVs in the same local genomics region but various alleles. It is extremely difficult to make correct calls for such heterogeneous SVs. Precise analysis of read alignments is needed to distinguish the signatures of the multiple SVs in such loci.

cuteSV is a novel SV detection fulfilling the requirements mentioned above. Using aligned reads (sorted BAM file) as input, it uses several specifically designed methods to extract the SV signatures of various kinds of SVs from aligned reads, and a clustering-and-refinement method to thoroughly analyze the collected signatures to call SVs. The approach has three major steps as following (a schematic illustration is in Figure 1).

1) cuteSV uses a series of signature extraction methods tailored for various types of SVs. This is one of the two cores of cuteSV which enables to comprehensively collect SV signatures and effectively recover the evidences of SVs from the complicated and fragile alignments.

2) cuteSV uses a specifically designed cluster-and-refinement approach to cluster the chimerically aligned reads in local regions, and further refines the clusters to precisely distinguish the SV signatures from heterozygous SVs. This is another core of cuteSV which enables to well-handle heterogeneous breakpoints in read alignments, and robust to those multi-allelic heterozygous SVs.

3) cuteSV uses heuristic rules to make SV calls based on the clustered SV signatures, and optionally perform SV genotyping.

cuteSV also implements a block division-based approach to process the input data in a parallel way with multiple CPU threads. Also refer to Materials and methods for more detailed information about the implementation of cuteSV.

The following sub-sections are mainly the benchmark results of cuteSV on various sequencing datasets, and more insights on the features of cuteSV are given in Discussion section.

SV detection with the HG002 PacBio data

We implemented cuteSV and three state-of-the-art long read-based SV callers, i.e. Sniffles, PBSV, and SVIM, on a 69× HG002 PacBio CLR dataset [36] (mean read length: 7938 bp) at first. PBMM2 was employed for read alignment. Another state-of-the-art aligner, NGMLR, was also employed to investigate the effect of the aligners to SV calling. The results mentioned are based on the alignments produced by PBMM2 if not specifically mentioned. A high-confidence insertion and deletion callset for this sample made by Genome in a Bottle Consortium (GIAB) [37] was employed as ground truth. Truvari (https://github.com/spiralgenetics/truvari) was used to assess the precision, recall, and F-measure (F1) of the callsets produced by various approaches.

The yields of the SV callers are shown in Figure 2A. In terms of the 69× CLR data, cuteSV simultaneously achieved the highest precision, recall, and F1, and all of them surpassed 94% in absolute term which is feasible to practical uses. The F1 of Sniffles (90.22%) and PBSV (91.48%) were comparable to each other, and both of them were slightly lower than that of cuteSV mainly due to their lower recall statistics (i.e. 86.27% and 88.42% for Sniffles and PBSV, respectively). SVIM achieved a fairly high recall (89.30%) with a relatively poor precision (75.13%), so that its F1 (81.60%) was lowest.

We further down-sampled the original dataset to 5X, 10X, 20X, 30X and 40X to assess the ability of the SV callers on lower coverage datasets. cuteSV still achieved higher recalls and F1s on all these datasets, indicating that it has outstanding sensitivity without loss of accuracy. It is also worth noting that cuteSV simultaneously achieved >90% precision, recall and F1 at 20X. This is beneficial to practical use, since some more cost-effective sequencing plans can be considered with this result. However, this is still hard for other callers since their recalls and F1s were much lower on the down-sampled datasets.

Besides PacBio CLR data, we also assessed the ability of the callers on the newly promoted PacBio CCS dataset of the same sample [38] [39] (coverage: 28X, mean read length: 13478 bp). The results of cuteSV and PBSV were very close to each other (Figure 2B) (precision: 95.2%, recall: 98.0%, F1: 96.6%), and they outperformed Sniffles and SVIM by 1% to 4% on various statistics. We also down-sampled the dataset to 5X and 10X to further assess the callers. On the 10X down-sampled dataset, PBSV achieved the highest recall (96.70%) and cuteSV was the best runner-up (93.79%). The slightly lower recall of cuteSV is mainly due to the setting of minimal signature size for processing in cuteSV (30 bp in default) was longer than it in PBSV (20 bp in default) exactly, which disabled cuteSV to identify more SVs using evidence in a shorter size. Moreover, it is also observed that, 5X coverage is enough to achieve >90% precision, recall and F1, for both of cuteSV and PBSV. Such results suggest that, with the higher sequencing quality, the requirement on data size becomes lower and more economical sequencing strategies could be feasible in genomics studies.

We further used GIAB Ashkenazi Trio callsets (HG002, HG003 and HG004) to calculate the Mendelian-Discordance-Rate (MDR) of the long read-based SV callers, to more comprehensively assess their ability to the detection of various types of SVs. The results (Figure 2C) suggest that the MDR of cuteSV is lowest for almost all types of SVs, i.e., DEL: 0.103, INS/DUP: 0.162, INV: 0.061, BND: 0.127, All Types: 0.131, respectively, indicating that the callset produced by cuteSV is more plausible. The MDR of Sniffles is higher but also comparable (All Types: 0.173), while the MDRs of PBSV (All Types: 0.590) and SVIM (All Types: 0.240) are much higher, indicating that their callsets could be less accurate. It is worth noting that PBSV and SVIM had high MDRs on translocations (0.72) and inversions (0.82), respectively, indicating that they could not well-handle such types of SVs.

SV detection with the HG002 ONT PromethION data

We further assessed the yields of the SV callers on a newly published ONT PromethION dataset of the HG002 human sample (mean read length: 17335 bp, coverage: 47x, available at ftp://ftp.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/UCSC_Ultralong_OxfordNanopore_Promethion/), which is aligned with Minimap2. The benchmark results (Figure 3A) indicate that, cuteSV achieved highest the precision (i.e. 92.40%), recall (i.e. 96.61%) and F1 (i.e. 94.46%) simultaneously, and its outperformance on Sniffles and SVIM were more obvious than that of the PacBio CLR dataset. It is also worth noting that PBSV crashed for this dataset. On the down-sampled datasets (5X, 10X and 20X), the outperformance of cuteSV was also remarkable. Particularly, cuteSV found most (85%) of ground truth SVs only at 10x coverage with high precision (93.76%) and F1-measure (89.16%). However, at the same coverage, the yields of Sniffles, SVIM and PBSV were much lower, i.e., their F1-measures were at least 9% lower than that of cuteSV.

We further compared the SV callsets produced by cuteSV based on HG002 PacBio CLR and ONT datasets. cuteSV called 26605 SVs from the PacBio dataset and 26427 SVs from the ONT dataset (see Supplementary Table 1), respectively. 7293 SVs (27.41%) of the PacBio callset and 7115 SVs (26.92%) of the ONT callset are unique. Moreover, in the 7293 PacBio-only calls, 54.15% (3949 of 7293) and 21.20% (1546 of 7293) are respectively insertions and deletions, and in the 7115 ONT-only calls, the corresponding numbers are respectively 47.94% (3411 of 7115) and 41.00% (2917 of 7115). Previous studies indicate that there are usually more false positive insertion and deletion calls made from PacBio datasets [23] and ONT datasets [24], respectively. Thus, the fractions of the unique calls from the PacBio and ONT datasets are plausible. More than that, it is observed that there are some SVs can only be discovered with the ONT dataset, mainly due to the larger read lengths. An example is shown in Supplementary Figure 1, that a 6481 bp insertion (breakpoint at chr1:9683994) is only detected with the ONT reads, since the ONT reads spanning this event are longer than that of the PacBio reads aligned to this locus, and a large proportion of the ONT reads carry significant insertion signals in their CIGARs.

Moreover, there were 33751 SV calls existing in both of the PacBio- and ONT-based callsets, and they covered 98.64% (9510 of 9641) of the ground truth callset of HG002 (Figure 3B), indicating a higher sensitivity than all the callsets made by the SV callers with a single dataset. This suggests that the combination of multiple datasets is feasible to produce high quality SV callset.

The performance of the SV callers

We assessed the speed (Figure 4A) and memory footprint (Figure 4B) of the SV callers on the 69X HG002 CLR dataset. The speed of the benchmarked approaches with a single CPU thread is comparable, i.e., Sniffles used 367 minutes which is the fastest one and followed by SVIM (423 minutes), cuteSV (983 minutes) and PBSV (1715 minutes). The relatively slower speed of cuteSV is mainly because of that it uses multiple methods to extract the signatures of SVs. We also asked the SV callers to run with multiple CPU threads, however, it was observed that none of Sniffles and PBSV had an obvious speedup with more CPU threads, i.e., the time costs were nearly the same to that of single threads, and SVIM does not support multiple-threading computing. This is possibly due to their implementations. However, cuteSV showed a nearly linear speedup with the number of CPU threads, and the wall clock time greatly reduced. For example, with 16 CPU threads, cuteSV used only 82 minutes to process the whole dataset. Moreover, the memory footprint of cuteSV (0.35GB) was smaller than that of other approaches by two order of magnitudes (i.e., Sniffles: 17.61GB, PBSV: 10.57GB and SVIM: 9.41GB). With its quasi linear multiple-threads speedup and low memory footprint, we realize that cuteSV is a highly scalable SV

detection tool, which is suited to high performance computing platforms and large-scale data analysis tasks such as the SV detection of many samples.

The effect of various read aligners for SV calling

To investigate the effect of various aligners to SV detection, we employed two aligners, PBMM2 and NGMLR, to separately align the reads of the 69X HG002 PacBio CLR dataset, and used their alignments as inputs to the four SV callers to produce various callsets. The callsets were then evaluated, and the effects of the aligners on the sensitivity and accuracy of SV detection were observed. Some details are as following.

1) We collected SV calls marked as false-negative (FN) by Truvari and compared them to investigate the effect of the two aligners on the sensitivity of the SV callers. The difference between the callsets produced by same SV callers with the alignments of different aligners are in Figure 5A and Supplementary Table 2. The results indicate that overall the SV callers had higher sensitivities (fewer FN calls) on insertions and deletions less than 1,000 bp with the alignments produced by PBMM2. Moreover, with PBMM2, PBSV discovered 436 more insertions than that of NGMLR, whose lengths are within 200 bp to 700 bp and 369 of them coincided with the sequences of Alu family. This issue is quite worth noting since MEI is a major category of SVs. On the other hand, cuteSV, Sniffles and PBSV respectively discovered 109, 283 and 127 more larger SVs (size >1000 bp) with the alignments of NGMLR, comparing to their own callsets based on the alignments of PBMM2, which is also worth noting. Meanwhile, the sensitivity of SVIM did not changed much with the alignments of the two aligners.

2) We further investigated the false-positive (FP) calls of the SV callers based on various aligners (see Figure 5B and Supplementary Table 3). The callsets of cuteSV, Sniffles and PBSV with PBMM2 had slightly higher numbers (148, 65 and 31, respectively) of <1000 bp FP insertions and deletions than that of NGMLR. However, the callset of SVIM with NGMLR had a large number (10578) of FP insertions, indicating that they could be not compatible. For >1000 bp SVs, the numbers of FPs were very close for the corresponding callsets of cuteSV, Sniffles and PBSV, however, there were still 939 more FPs in the callset of SVIM with NGMLR than that of PBMM2.

3) The MDRs of the callsets with various aligners were also assessed (Figure 5C). It was observed that cuteSV, Sniffles and PBSV had decreased MDRs (by 0.35% to 7.52%) on deletions and translocations, but no significant difference was observed for insertions/duplications and inversions with the two aligners. Moreover, overall SVIM had obviously higher MDRs with NGMLR, indicating that PBMM2 could be better choice for it.

The results of cuteSV with various configurations of parameters

We assessed the yields of cuteSV with various configurations of two critical parameters, –min_support and –min_size (-s and -l in the software, respectively), on various coverages (5×, 10×, 20×, 30×, 40× and 69×) of HG002 CLR datasets.

The –min_support parameter indicates the minimal number of supporting reads to make an SV call. As the results shown in Figure 6A and Supplementary Table 4, with the default setting of –min_size (–min_size=30), cuteSV achieved best yields when –min_support was configured as 1 to 10 for the various coverages, and there is an obvious trade-off between precision and recall, i.e., setting -min_support with smaller number might result in higher sensitivity as well as lower precision and vice versa.

The –min_size parameter indicates the minimal size of SV signature considered in clustering. Keeping –min_support parameter fixed, we investigated the results of cuteSV with two different settings on -

min_size (i.e., -min_size=30 and -min_size=50, Figure 6B). At various coverages, the accuracies of cuteSV were 0.38% to 1.71% higher with the setting of –min_size=50, while the recall rates were 0.06% to 2.19% higher with –min_size=30. That indicates that setting -min_size with smaller numbers might result in higher sensitivity but lower accuracy, and vice versa. It is also worthnoting that, although the trade-off exists, for each of the coverages, the F-measures of cuteSV with various settings are quite close to each other (the difference is less than 1%).

SV detection with the NA19240 PacBio CLR data

A PacBio CLR dataset from another well-studied human sample (NA19240) [40] was also employed (mean read length: 6503 bp, coverage: 40x) to more comprehensively benchmark the SV callers. A callset for this sample [41] was used as ground truth.

The precision, recall, and F1-measure of the benchmarked SV callers are in Figure 7 and Supplementary Table 5. cuteSV had the highest recall rates for all types of SVs (i.e., DEL: 62.79%, INS/DUP: 55.68%, INV: 19.11% and All Types: 58.64%), and in absolute terms cuteSV successfully discovered 1016 more SVs than the best runner-up (208 deletions, 768 insertions/duplications and 49 inversions). Nonetheless, the precision of cuteSV is lower, but still comparable to PBSV (56.71% vs 59.93% for All Types). Overall, cuteSV achieved the highest F1-measures (57.66% for All Types).

Discussion

Long read sequencing technologies provide the opportunity to comprehensive SV discovery, however, it is still non-trivial to exert the advantage of long reads due to the complicated alignments. In this article, we propose cuteSV, a sensitive, fast and scalable long read-based SV detection approach. Mainly, we show how to use a series of signature extraction methods and a tailored clustering-and-refinement method to implement a precise analysis of SV signatures to achieve good yields and performance simultaneously. Mainly, the cuteSV approach has four major contributions as following.

1) Read aligners could produce fragile alignments around SV events, for example, some reads could have several smaller insertions and/or deletions in their alignments around a larger SV event. Such alignments usually mislead SV detection approaches to make incorrect calls. However, by some specifically designed methods, cuteSV enables to adaptively combine the chimeric alignments in reads to effectively recover the signatures of the real SV events. An example is in Supplementary Figure 2, which is an 272bp insertion event successfully discovered by cuteSV, but none of other benchmarked SV callers, potentially due to the fragile alignments.

2) There are multi-allelic heterozygous SVs in some local regions, and it is even more difficult that the multiple alternative alleles have the same SV type. Since most of state-of-the-art SV callers rely on genomic position-based clustering of chimeric reads, the reads spanning various SVs in the local region could be binned in one cluster, and then incorrect SV calls are made. cuteSV solves this problem by using a tailored cluster refinement approach, which precisely analyzes the SV signatures of the reads in same clusters. This operation enables to distinguish the reads spanning different SV alleles, and correctly bins them by various sub-clusters, which greatly helps the detection of such complex SV events. Two examples are shown in Supplementary Figure 3, one is a heterozygous insertion event (a 180bp and a 36bp insertions in the same region) and the other one is a heterozygous deletion event (a 123bp and a 37 bp deletions in the same region). Only cuteSV and PBSV successfully discovered them in the benchmark.

3) With its tailored signature extraction and clustering methods, cuteSV enables to more sensitively detect SVs. The advantage is more obvious for relatively low coverage datasets. The benchmark results

suggest that cuteSV can discover most of SVs with 20X coverage datasets for human samples, without loss of accuracy. This is helpful to make more flexible sequencing plans in large-scale genomic studies, e.g., population genomics studies, to achieve the goals in a more economical way.

4) Most of state-of-the-art SV callers are still not very scalable, i.e., they cannot fully take the advantage of computational resources to speedup. This could become a problem for many large datasets since more computational nodes are needed and/or the time cost might be very high. However, cuteSV has good scalability with its block division implementation, which enables to process the input dataset in a parallel way and achieve a nearly linear speedup with the number of CPU threads. This makes it especially suited to modern HPC resources and being helpful to upcoming large-scale genomics studies.

The benchmark results also indicate that there are still some SVs cannot be successfully detected by cuteSV. We investigated the intermediate results of cuteSV and found that most of the false negative calls are due to that the read alignments are not informative enough or inaccurate. Two typical examples are shown in Supplementary Figure 4 and 5. In Supplementary Figure 4, a 98bp deletion case is shown that deletion signatures emerge in nearly all the reads around the event, however, the size of the deletions in the alignment are not correct, i.e., most of them are around 50bp. In Supplementary Figure 5, a 707bp insertion case is shown that, insertion signatures also emerge and their sizes are close to the SV event, however, the positions of the breakpoints in the reads are quite far from the ground truth breakpoint. Under such circumstance, all the benchmarked SV callers made SV calls, however, the sizes and/or positions are incorrect, since the read alignments are misleading. We realize that it could be still an important future work to develop more accurate long read aligners.

Conclusion

Long-read sequencing technologies provide the opportunity to build a comprehensive map of structural variations of human genome. However, there are still technical issues to be addressed to further improve the sensitivity, accuracy and performance of long read-based SV detection, and the development of advanced computational approaches are on demand.

In this article, we provide a novel long read-based SV detection approach, cuteSV. With its stepwise refinement clustering algorithm, it enables to thoroughly analyze the complex signatures of SVs implied by inter- and intra-alignment to achieve. Benchmark results demonstrate that cuteSV achieves good yields and performance simultaneously. Moreover, it has outstanding ability to detect SVs with low coverage sequencing data and high scalability to handle large-scale datasets. We believe that cuteSV will have many potentials in cutting-edge genomics studies to fully exert the advantages of long read sequencing data.

Materials and methods

Read alignment

cuteSV uses sorted BAM files as input, and it supports to employ state-of-the-art long read aligners to compose SV detection pipeline. Aligners having good ability to well-handle the large insertions/deletions in reads and/or produce accurate split alignments are preferable, since cuteSV extracts important SV signatures from such alignments. In the Results section, it demonstrates that state-of-the-art aligners like PBMM2 and NGMLR are feasible to the cuteSV approach.

Extraction of SV signatures implied by CIGARs

Given a set of aligned reads, cuteSV separately analyzes the detailed alignment of each read. Mainly, it extracts two categories of SV signatures, implied by the long insertions/deletions in CIGARs and split alignments, respectively. In the cuteSV approach, the signatures of SVs are represented as a 3-tuple, i.e., $SIG = (pos, len, ID)$, where pos indicates the starting position on the reference genome, len indicates the size of SV and ID indicates the unique read ID. cuteSV composes the SV signals implied by read alignments into $SIGs$, and $SIGs$ of various reads are clustered and analyzed to call SVs.

cuteSV uses a precise method to extract and merge the long insertions/deletions in CIGARs to transform them to SV signatures. This method enables to recover the SV signatures of long insertions/deletions which are initially partitioned as multiple trivial indels by read aligners, and it helps the inference of the breakpoints and lengths of SVs. In details, cuteSV extracts the insertion/deletions having >30 bp size described by the CIGARs of the reads, and composes them into $SIGs$ with their positions, lengths and read IDs. For two signatures, SIG_1 and SIG_2, cuteSV merges them if they meet the following condition:

$$\begin{cases} pos_2 - pos_1 \leq S_{dis} & if\ INS \\ pos_2 - (pos_1 + len_1) \leq S_{dis} & if\ DEL \end{cases} \quad (1)$$

where S_{dis} is a threshold on the distance between the two signatures (default value: 500 bp). The two signatures are then merged as $SIG_M = (pos_1, len_1 + len_2, ID)$. The merging operation enhances the signatures of large insertions/deletions, and all the remaining signatures after merging are used as informative signatures for further processing.

Extraction of SV signatures implied by split alignments

For a read having split alignment(s) (described by its primary and supplementary alignments), cuteSV records each split by a 6-tuple (also termed as a "segment"), $Seg = (read_s, read_e, Ref_s, Ref_e, Chr, Strand)$, where $read_s$, $read_e$, Ref_s, Ref_e respectively indicate the starting and end coordinates on read and reference genome, and Chr and $Strand$ respectively indicate its chromosome and orientation. cuteSV uses several heuristic rules as following to recover SV signatures from these $Segs$.

(1) Extraction of deletions/insertions signatures. If two segments, Seg_1 and Seg_2, are adjacent on the read and aligned to the same chromosome with identical orientations, cuteSV computes $Diff_{dis} = (Ref_{2s} - Ref_{1e}) - (read_{2s} - read_{1e})$ and $Diff_{olp} = Ref_{1e} - Ref_{2s}$. If $Diff_{olp} < 30\ bp$ and $Diff_{dis} \geq 30\ bp$, cuteSV considers that the two segments indicate a deletion event, and compose a deletion signature:

$$SIG_{DEL} = (Ref_{1e}, Diff_{dis}, ID) \quad (2)$$

Moreover, if $Diff_{olp} < 30\ bp$ and $Diff_{dis} \leq -30\ bp$, cuteSV considers that there is an insertion event, and compose an insertion signature:

$$SIG_{INS} = \left(\frac{Ref_{1e} + Ref_{2s}}{2}, -Diff_{dis}, ID\right) \quad (3)$$

(2) Extraction of duplication signatures. If two adjacent segments are mapped to similar positions, i.e., $Diff_{olp} \geq 30\ bp$, cuteSV composes a duplication signature:

$$SIG_{DUP} = (Ref_{2s}, Ref_{1e}, ID) \quad (4)$$

(3) Extraction of inversion signatures. If two adjacent segments are mapped to same chromosome but different strands, cuteSV composes an inversion signature:

$$SIG_{INV} = \begin{cases} \begin{matrix}(Ref_{1e}, Ref_{2e}) \\ (Ref_{1s}, Ref_{2s})\end{matrix}, if\ LEN_{INV} \geq \quad 50\ bp \\ \begin{matrix}(Ref_{2e}, Ref_{1e}) \\ (Ref_{2s}, Ref_{1s})\end{matrix}, if\ LEN_{INV} \leq -50\ bp \end{cases} \tag{5}$$

where $LEN_{INV} = Ref_{2e} - Ref_{1e}$ indicates the length of the event (with direction). It is also worth noting that the primary strand of the inversion is unknown, so that cuteSV records all possible situations.

(4) Extraction of translocation signatures. If two adjacent segments are mapped to different chromosomes, and the two segments are distant <100bp on the reads, cuteSV composes a translocation signature:

$$SIG_{BND} = \begin{cases} (Chr_1, Ref_{1e}, Chr_2, Ref_{2s}, ID) & , if\ "0 \leq " \\ (Chr_2, Ref_{2s}, Chr_1, Ref_{1e}, ID) & , if\ "0 \geq " \\ (Chr_1, Ref_{1e}, Chr_2, Ref_{2e}, ID) & , if\ "1 \leq " \\ (Chr_2, Ref_{2e}, Chr_1, Ref_{1e}, ID) & , if\ "1 \geq " \\ (Chr_1, Ref_{1s}, Chr_2, Ref_{2s}, ID) & , if\ "2 \leq " \\ (Chr_2, Ref_{2s}, Chr_1, Ref_{1s}, ID) & , if\ "2 \geq " \\ (Chr_1, Ref_{1s}, Chr_2, Ref_{2e}, ID) & , if\ "3 \leq " \\ (Chr_2, Ref_{2e}, Chr_1, Ref_{1s}, ID) & , if\ "3 \geq " \end{cases} \tag{6}$$

where 0, 1, 2, 3 indicates the combination of strands "++", "+-", "-+", "--", respectively, "≤" and "≥" means the chromosome ID of Chr_1 is in front of that of Chr_2 and vice versa.

(5) cuteSV uses a specifically designed method to extract the signatures of a kind of complex SVs, that there is a mobile insertion in between two duplicated sequences. An example is in Supplementary Figure 6, that there are two duplicated local sequences (both of their mapped positions are around Chr1:73594981), and there is another local sequence within them in the read (whose mapped position is in a decoy sequence of hs37d5). cuteSV extracts a series of signatures, including a duplication (as shown in Eq.4), a translocation (as shown in Eq.6) and an insertion (as shown in Eq.7), to describe such complex SV events.

$$SIG_{INS} = (\min(Ref_{1e}, Ref_{2s}), -Diff_{dis}, ID) \tag{7}$$

Clustering of SV signatures

cuteSV clusters the collected SV signatures (SIGs) in two-step approach. The first step clusters the signatures by their genomic positions and types to bin the signatures in various local regions, and the second step clusters the signatures by their length to distingue the signatures in similar regions but belonging to heterozygous SVs (i.e., both of the two alleles are being but different SVs in such loci). Some details are as following

In the first step, cuteSV sorts all the SV signatures by their genomic positions and types (i.e., deletions, insertions, duplications, inversions and translocations) at first. For each of the category, cuteSV initially creates a new cluster, and scans all the signatures from upstream to downstream to add them into the cluster in an iterative approach. More precisely, for a newly scanned SV signature SIG_i, cuteSV adds it into the cluster if there has been at least one signature SIG_j in the cluster which meets the following condition:

$$SIG_{\mathrm{i}}(pos) - SIG_{\mathrm{j}}(pos) \leq TH_{type}$$

$$\text{or } \left|\left(SIG_{\mathrm{i}}(pos) + SIG_{\mathrm{i}}(len)\right) - \left(SIG_{\mathrm{j}}(pos) + SIG_{\mathrm{j}}(len)\right)\right| \leq TH_{type} \quad (8)$$

where TH_{type} is a threshold on the distances among the clustered signatures, and different values are used for various types of SVs (TH_{type} is set configured as 50 to 500 bp, typically). If SIG_i cannot be added into the cluster, cuteSV creates a new cluster only having SIG_i and goes to next SV signature.

In the second step, cuteSV initially checks the numbers of signatures of the generated clusters, and discards the clusters having too few signatures. The remaining clusters are then refined by the following methods according to their SV types.

1) Refinement of deletion/insertion clusters. Given a deletion or insertion cluster, cuteSV sorts all its signatures by their sizes, and computes a parameter, Len_{bias}, with the following equation:

$$Len_{bias} = \alpha \times \frac{1}{|Group_{SIG}|} \sum_{1 \leq k \leq |Group_{SIG}|} Size_k \quad (9)$$

where α is a weighting parameter, $|Group_{SIG}|$ is the number of the signatures in the cluster, and $Size_k$ is the size of the k-th longest signature in $Group_{SIG}$. The configuration of α is related to the SV type, i.e., the default value of α is 0.2 for an insertion cluster and high error rate reads (e.g., PacBio CLR and ONT reads), and $\alpha = 0.65$ for low error rate reads (e.g., PacBio CCS reads), moreover, $\alpha = 0.3$ for a deletion cluster (regardless of error rate).

With Len_{bias}, cuteSV divides the cluster into sub-clusters which each of the sub-clusters is a potential SV allele in a local genomic region. In details, cuteSV initially creates one sub-cluster and add the signatures with largest size into the sub-cluster. Then cuteSV iteratively scans the signatures by the order of their sizes (from largest to smallest), i.e., for a newly scanned signature, if the difference between its size and the size of the last signature being added to the sub-cluster is smaller than Len_{bias}, cuteSV adds the newly scanned one into the sub-cluster; otherwise, a new sub-cluster is created with the newly scanned signature.

cuteSV recognizes the generated sub-clusters with highest number of signatures as a "major allele" sub-cluster if it meets the following condition:

$$SupR \geq SR_{min} \text{ and } SupR > \mu \times |Group_{SIG}| \quad (10)$$

where $SupR$ and SR_{min} are the number of its supporting reads and the threshold on the minimum number supporting reads, respectively. μ is a weighting parameter, the default values for an insertion cluster are 0.6 and 0.65 when using PacBio CLR/ONT reads and PacBio CCS reads, respectively. For a deletion cluster, the default values are 0.7 and 0.35 for the dataset mentioned above, respectively. If a "major allele" sub-cluster exists, cuteSV recognizes each of the remaining sub-clusters as a "minor allele" sub-cluster if it has more than SR_{min} signatures.

However, there is occasionally no "major allele" sub-cluster existing due to lack of enough supporting reads. cuteSV uses another heuristic rule that it recognizes the two largest sub-clusters $SupR_{first}$ and $SupR_{second}$ as "major allele" and "minor allele" sub-cluster it they meet the following condition:

$$\begin{cases} SR_{min} \leq SupR_{first} \leq \mu \times |Group_{SIG}| \\ 0.4 \times |Group_{SIG}| \leq SupR_{second} \leq SR_{min} \\ 0.95 \times |Group_{SIG}| \leq SupR_{first} + SupR_{second} \end{cases} \quad (11)$$

This rule indicates that almost all of the SV signatures support the two alleles, meanwhile, both of them occupy >40% supporting signatures of this cluster.

2) Refinement of duplication/inversion clusters. Given a duplication or inversion cluster, cuteSV initially creates one or more sub-clusters that the breakpoints of all the signatures in the same sub-cluster are within 500bp. Further, cuteSV recognizes the "major allele" and "minor allele" sub-clusters with a heuristic rule similar to Eq.10 but sets $\mu = 1/3$ for duplication and $\mu = 1/3$ for inversion.

(3) Refinement of translocation clusters. Given a translocation cluster, cuteSV initially creates one or more sub-clusters that the breakpoints of all the signatures in the same sub-cluster are within 50bp. Further, cuteSV also recognizes the "major allele" and "minor allele" sub-clusters with a heuristic rule similar to Eq.10 but sets $\mu = 0.6$ and SR_{min} as half of the value used for deletion/insertion clusters. This setting mainly considers the diverse combinations of chromosome and orientation of translocation events and is beneficial to the sensitivity.

SV calling and genotyping

For each cluster of signatures, cuteSV computes the weighted average of the positions and sizes to predict the breakpoint(s) and size of the corresponding SV. Further, it removes the predicted SVs being <30 bp in size.

Moreover, cuteSV supports to implement SV genotyping (which is an optional step in the tool). In details, cuteSV uses a local genomic region for a predicted SV to analyze the ratio of supporting reads, which is defined as:

$$Area_{at} = \left(\max(BK_1 - bias, 0), \min(BK_2 + bias, L_{REF})\right) \quad (18)$$

where BK_1 is the first breakpoint, BK_2 is the next breakpoint, $bias$ means the scope of extension (20 bp, in default), and L_{REF} indicates the maximum length of chromosome took into consideration. cuteSV counts the numbers of all the reads and the reads supporting the predicted SV within $Area_{at}$. Various genotypes are assigned by cuteSV according the ratio of supporting reads, i.e., homozygous reference allele (0/0) for a ratio ≤ 0.3, homozygous variant allele (1/1) for a ratio ≥ 0.8, and heterozygous variant allele (0/1) for a ratio in between.

Implementation of long-read mapping and SV calling

We used PBMM2 (version 1.0.0), NGMLR (version 0.2.3) and Minimap2 (version 2.17) to implement the read alignment of the benchmarking datasets (Supplementary Table 6). The parameter "--preset" of long-read aligners were tuned for different sequencing technologies. Samtools (version 1.9) was employed for read extraction, sorted BAM generation, and sequencing data down-sampling.

Sniffles (version 1.0.11), PBSV (version 2.2.0), SVIM (version 0.4.3) and cuteSV (version 1.0.1) were benchmarked with the sorted BAM files as input. Specifically, for Sniffles, the configurations "-l 50 -s 2/3/3/4/5/10" were used for PacBio CLR datasets, "-l 30 -s 1/2/3" for PacBio CCS reads, and "-l 30 -s 2/3/4/10" for ONT PromethION reads, respectively. For PBSV, default settings were used for PacBio CLR and ONT PromethION data, and "--preset CCS" for PacBio CCS data. For SVIM, the configuration "--cluster_max_distance 1.4" was employed for all datasets in this study. For cuteSV, the configuration "-l 30 --max_cluster_bias_INS 100 --diff_ratio_merging_INS 0.2 --diff_ratio_filtering_INS 0.6 --diff_ratio_filtering_DEL0.7" was used for PacBio CLR ("-s 2/3/3/4/5/10") and ONT PromethION reads ("-s 2/3/4/10"), and "-l 30 --max_cluster_bias_INS 200 --diff_ratio_merging_INS 0.65 --diff_ratio_filtering_INS 0.65 --diff_ratio_filtering_DEL0.35 -s 1/2/3" was used for PacBio CCS reads.

Evaluation of SV callsets

The evaluation of the HG002 human sample is run with Truvari (version 1.1), and the high confidence insertion and deletion callsets (version 0.6) are used as gold standard sets. Before evaluation, we preprocess the SV calls of each tool. For Sniffles, we discard inversions and translocations, and transform duplications to insertions. For SVIM, we delete SV calls that less than 40 in the quality score as author recommended, and transform duplications to insertions as well. For PBSV and cuteSV, we only select insertions and deletions for assessment. Then, BGZIP and TABIX are employed to compress and index the processed VCF files.

In evaluation, only SV calls belonging to 50 bp to 10 kbp in size, and contained within the GIAB high confidence regions (version 0.6) are considered for evaluating. A prediction is determined as a true-positive (TP) when meeting

$$\begin{cases} \max(comp_s - 1kbp, base_s) \leq \min(comp_e + 1kbp, base_e) \\ \min(comp_{size} + base_{size})/\max(comp_{size} + base_{size}) \geq 0.7 \\ comp_{type} = base_{type} \end{cases} \tag{19}$$

where $comp_s$, $comp_e$, $comp_{size}$ and $comp_{type}$ indicate start coordinate, stop coordinate, size and SV class of a prediction, and $base_s$, $base_e$, $base_{size}$ and $base_{type}$ are also starting coordinate, stopping coordinate, size and SV class of a call in truth set, respectively. On the contrary, a prediction is determined as a false-positive (FP) if unsatisfied Eq.19. And a false-negative (FN) is assigned when there is a base call that cannot be detected by any SV calls.

Based on the results above, Precision or the ratio of TPs to total calls in predictions is defined as

$$Precision = \frac{TPs}{TPs+FPs} \tag{20}$$

Similarly, Recall or the ratio of TPs to total calls in truth set is defined as

$$Recall = \frac{TPs}{TPs+FNs} \tag{21}$$

F-measure (F1 score), a measurement of weighted averaging of both precision and recall, which is

$$F_measure = \frac{2 \times Precision \times Recall}{Precision+Recall} \tag{22}$$

The evaluation of the NA19240 is similar to the HG002 benchmarking. We adopt Eq.19 to assess every deletion, insertion (duplication regarded as the subset of insertion) and inversion against the callsets generated from the study [41]. And then applied Eq.20 to 22 for summarizing the performance on SV calling.

For the evaluation of the Ashkenazi trio, in order to measure the performance of an SV caller via the pedigree between offspring and its parents, the callsets of parents (i.e. HG003 and HG004) produced by corresponding methods are adopted as truth sets. For each SV call of the son (i.e. HG002) except translocations, we use Eq.19 to assign true-positive calls. As an arbitrary rearrangement type, a translocation or BND is considered precise when meeting

$$\begin{cases} |comp_{BK1} - base_{BK1}| \leq 1\text{kbp} \\ |comp_{BK2} - base_{BK2}| \leq 1\text{kbp} \\ comp_{chr1} = base_{chr1} \\ comp_{chr2} = base_{chr2} \end{cases} \tag{23}$$

where $BK1$, $BK2$, $chr1$, and $chr2$ are the combination of breakends and chromosomes of a call on the offspring and its parents, respectively. Therefore, the ratio of MDR can be reckoned via

$$MDR = \frac{\sum SVs\ in\ son - \sum precise\ SVs\ in\ son}{\sum SVs\ in\ son} \tag{24}$$

To evaluate the computational performance of the four SV callers under different threads, the runtime and memory using are assessed by "/usr/bin/time -v" command of Linux Operating System. In the output results of the command, "Elapsed (wall clock) time" and "Maximum resident set size" indicate the elapsed runtime and memory consumption, respectively. It is worth noting that SV calling performed by PBSV involves two steps, i.e. discover and call, we use the sum of wall clock time of both two steps as final elapsed runtime, whereas the memory consumption depends on the maximum memory using of the two runs.

For more details please see Supplementary Notes.

Authors' contributions

T.J. conceived and supervised the project. T.J. developed cuteSV. T.J., B.L. and Y.J. performed analysis. T.J. and B.L. wrote the manuscript. All authors read and approved the final manuscript.

Funding

This work has been supported by the National Key Research and Development Program of China (Nos: 2017YFC0907503, 2018YFC0910504, and 2017YFC1201201).

Availability of data and material

The human reference genome, raw sequencing data, alignments and gold standard truth callsets used in this study are available from the respective publications and websites listed in Supplementary Table 6. The implementation of cuteSV can be downloaded from GitHub (https://github.com/tjiangHIT/cuteSV).

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

References

1. Sudmant PH, Rausch T, Gardner EJ, Handsaker RE, Abyzov A, Huddleston J, Zhang Y, Ye K, Jun G, Fritz MH, et al:

An integrated map of structural variation in 2,504 human genomes. *Nature* 2015, **526:**75-81.

2. Alkan C, Coe BP, Eichler EE: **Genome structural variation discovery and genotyping.** *Nat Rev Genet* 2011, **12:**363-376.
3. Rovelet-Lecrux A, Hannequin D, Raux G, Le Meur N, Laquerriere A, Vital A, Dumanchin C, Feuillette S, Brice A, Vercelletto M, et al: **APP locus duplication causes autosomal dominant early-onset Alzheimer disease with cerebral amyloid angiopathy.** *Nature Genetics* 2006, **38:**24-26.
4. Hedges DJ, Hamilton-Nelson KL, Sacharow SJ, Nations L, Beecham GW, Kozhekbaeva ZM, Butler BL, Cukier HN, Whitehead PL, Ma DQ, et al: **Evidence of novel fine-scale structural variation at autism spectrum disorder candidate loci.** *Molecular Autism* 2012, **3**.
5. Weischenfeldt J, Symmons O, Spitz F, Korbel JO: **Phenotypic impact of genomic structural variation: insights from and for human disease.** *Nat Rev Genet* 2013, **14:**125-138.
6. Macintyre G, Ylstra B, Brenton JD: **Sequencing Structural Variants in Cancer for Precision Therapeutics.** *Trends in Genetics* 2016, **32:**530-542.
7. Dennenmoser S, Sedlazeck FJ, Iwaszkiewicz E, Li XY, Altmuller J, Nolte AW: **Copy number increases of transposable elements and protein-coding genes in an invasive fish of hybrid origin.** *Mol Ecol* 2017, **26:**4712-4724.
8. Lupski JR: **Structural variation mutagenesis of the human genome: Impact on disease and evolution.** *Environ Mol Mutagen* 2015, **56:**419-436.
9. Chiang C, Scott AJ, Davis JR, Tsang EK, Li X, Kim Y, Hadzic T, Damani FN, Ganel L, Consortium GT, et al: **The impact of structural variation on human gene expression.** *Nat Genet* 2017, **49:**692-699.
10. Zichner T, Garfield DA, Rausch T, Stutz AM, Cannavo E, Braun M, Furlong EEM, Korbel JO: **Impact of genomic structural variation in Drosophila melanogaster based on population-scale sequencing.** *Genome Research* 2013, **23:**568-579.
11. Jeffares DC, Jolly C, Hoti M, Speed D, Shaw L, Rallis C, Balloux F, Dessimoz C, Bahler J, Sedlazeck FJ: **Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast.** *Nature Communications* 2017, **8**.
12. Levy S, Sutton G, Ng PC, Feuk L, Halpern AL, Walenz BP, Axelrod N, Huang J, Kirkness EF, Denisov G, et al: **The diploid genome sequence of an individual human.** *PLoS Biol* 2007, **5:**e254.
13. Wheeler DA, Srinivasan M, Egholm M, Shen Y, Chen L, McGuire A, He W, Chen YJ, Makhijani V, Roth GT, et al: **The complete genome of an individual by massively parallel DNA sequencing.** *Nature* 2008, **452:**872-876.
14. Yoon ST, Xuan ZY, Makarov V, Ye K, Sebat J: **Sensitive and accurate detection of copy number variants using read depth of coverage.** *Genome Research* 2009, **19:**1586-1592.
15. Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC, Zhang QY, Locke DP, et al: **BreakDancer: an algorithm for high-resolution mapping of genomic structural variation.** *Nature Methods* 2009, **6:**677-+.
16. Ye K, Schulz MH, Long Q, Apweiler R, Ning ZM: **Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads.** *Bioinformatics* 2009, **25:**2865-2871.
17. Chen K, Chen L, Fan X, Wallis J, Ding L, Weinstock G: **TIGRA: A targeted iterative graph routing assembler for breakpoint assembly.** *Genome Research* 2014, **24:**310-317.
18. Hormozdiari F, Hajirasouliha I, Dao P, Hach F, Yorukoglu D, Alkan C, Eichler EE, Sahinalp SC: **Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery.** *Bioinformatics* 2010, **26:**i350-i357.
19. Jiang Y, Wang YD, Brudno M: **PRISM: Pair-read informed split-read mapping for base-pair level detection of insertion, deletion and structural variants.** *Bioinformatics* 2012, **28:**2576-2583.
20. Rausch T, Zichner T, Schlattl A, Stutz AM, Benes V, Korbel JO: **DELLY: structural variant discovery by integrated paired-end and split-read analysis.** *Bioinformatics* 2012, **28:**I333-I339.

21. English AC, Salerno WJ, Hampton OA, Gonzaga-Jauregui C, Ambreth S, Ritter DI, Beck CR, Davis CF, Dahdouli M, Ma S, et al: **Assessing structural variation in a personal genome-towards a human reference diploid genome.** *Bmc Genomics* 2015, **16**.

22. Tattini L, D'Aurizio R, Magi A: **Detection of Genomic Structural Variants from Next-Generation Sequencing Data.** *Front Bioeng Biotechnol* 2015, **3:**92.

23. Roberts RJ, Carneiro MO, Schatz MC: **The advantages of SMRT sequencing.** *Genome Biol* 2013, **14:**405.

24. Jain M, Olsen HE, Paten B, Akeson M: **The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community.** *Genome Biol* 2016, **17:**239.

25. Sedlazeck FJ, Lee H, Darby CA, Schatz MC: **Piercing the dark matter: bioinformatics of long-range sequencing and mapping.** *Nat Rev Genet* 2018, **19:**329-346.

26. Goodwin S, McPherson JD, McCombie WR: **Coming of age: ten years of next-generation sequencing technologies.** *Nat Rev Genet* 2016, **17:**333-351.

27. English AC, Salerno WJ, Reid JG: **PBHoney: identifying genomic variants via long-read discordance and interrupted mapping.** *BMC Bioinformatics* 2014, **15:**180.

28. Huddleston J, Chaisson MJP, Steinberg KM, Warren W, Hoekzema K, Gordon D, Graves-Lindsay TA, Munson KM, Kronenberg ZN, Vives L, et al: **Discovery and genotyping of structural variation from long-read haploid genome sequence data.** *Genome Research* 2017, **27:**677-685.

29. Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, Schatz MC: **Accurate detection of complex structural variations using single-molecule sequencing.** *Nature Methods* 2018, **15:**461-+.

30. Heller D, Vingron M: **SVIM: Structural Variant Identification using Mapped Long Reads.** *Bioinformatics* 2019.

31. Chaisson MJ, Tesler G: **Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory.** *Bmc Bioinformatics* 2012, **13**.

32. Li H: **Minimap2: pairwise alignment for nucleotide sequences.** *Bioinformatics* 2018, **34:**3094-3100.

33. Jiang T, Liu B, Li J, Wang Y: **rMETL: sensitive mobile element insertion detection with long read realignment.** *Bioinformatics* 2019, **35:**3484-3486.

34. Shao H, Ganesamoorthy D, Duarte T, Cao MD, Hoggart CJ, Coin LJM: **npInv: accurate detection and genotyping of inversions using long read sub-alignment.** *BMC Bioinformatics* 2017, **19:**261.

35. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Genome Project Data Processing S: **The Sequence Alignment/Map format and SAMtools.** *Bioinformatics* 2009, **25:**2078-2079.

36. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, Weng Z, Liu Y, Mason CE, Alexander N, et al: **Extensive sequencing of seven human genomes to characterize benchmark reference materials.** *Sci Data* 2016, **3:**160025.

37. Zook JM, Hansen NF, Olson ND, Chapman LM, Mullikin JC, Xiao C, Sherry S, Koren S, Phillippy AM, Boutros PC, et al: **A robust benchmark for germline structural variant detection.** *bioRxiv* 2019**:**664623.

38. Travers KJ, Chin CS, Rank DR, Eid JS, Turner SW: **A flexible and efficient template format for circular consensus sequencing and SNP detection.** *Nucleic Acids Research* 2010, **38**.

39. Wenger AM, Peluso P, Rowell WJ, Chang PC, Hall RJ, Concepcion GT, Ebler J, Fungtammasan A, Kolesnikov A, Olson ND, et al: **Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome.** *Nat Biotechnol* 2019.

40. Clarke L, Fairley S, Zheng-Bradley X, Streeter I, Perry E, Lowy E, Tasse AM, Flicek P: **The international Genome sample resource (IGSR): A worldwide collection of genome variation incorporating the 1000 Genomes Project data.** *Nucleic Acids Res* 2017, **45:**D854-D859.

41. Chaisson MJP, Sanders AD, Zhao X, Malhotra A, Porubsky D, Rausch T, Gardner EJ, Rodriguez OL, Guo L, Collins RL, et al: **Multi-platform discovery of haplotype-resolved structural variation in human genomes.** *Nat Commun* 2019, **10:**1784.

42. Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, Hide W, Salit M: **Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls.** *Nat Biotechnol* 2014, **32:**246-251.

43. Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, Tyson JR, Beggs AD, Dilthey AT, Fiddes IT, et al: **Nanopore sequencing and assembly of a human genome with ultra-long reads.** *Nat Biotechnol* 2018, **36:**338-345.

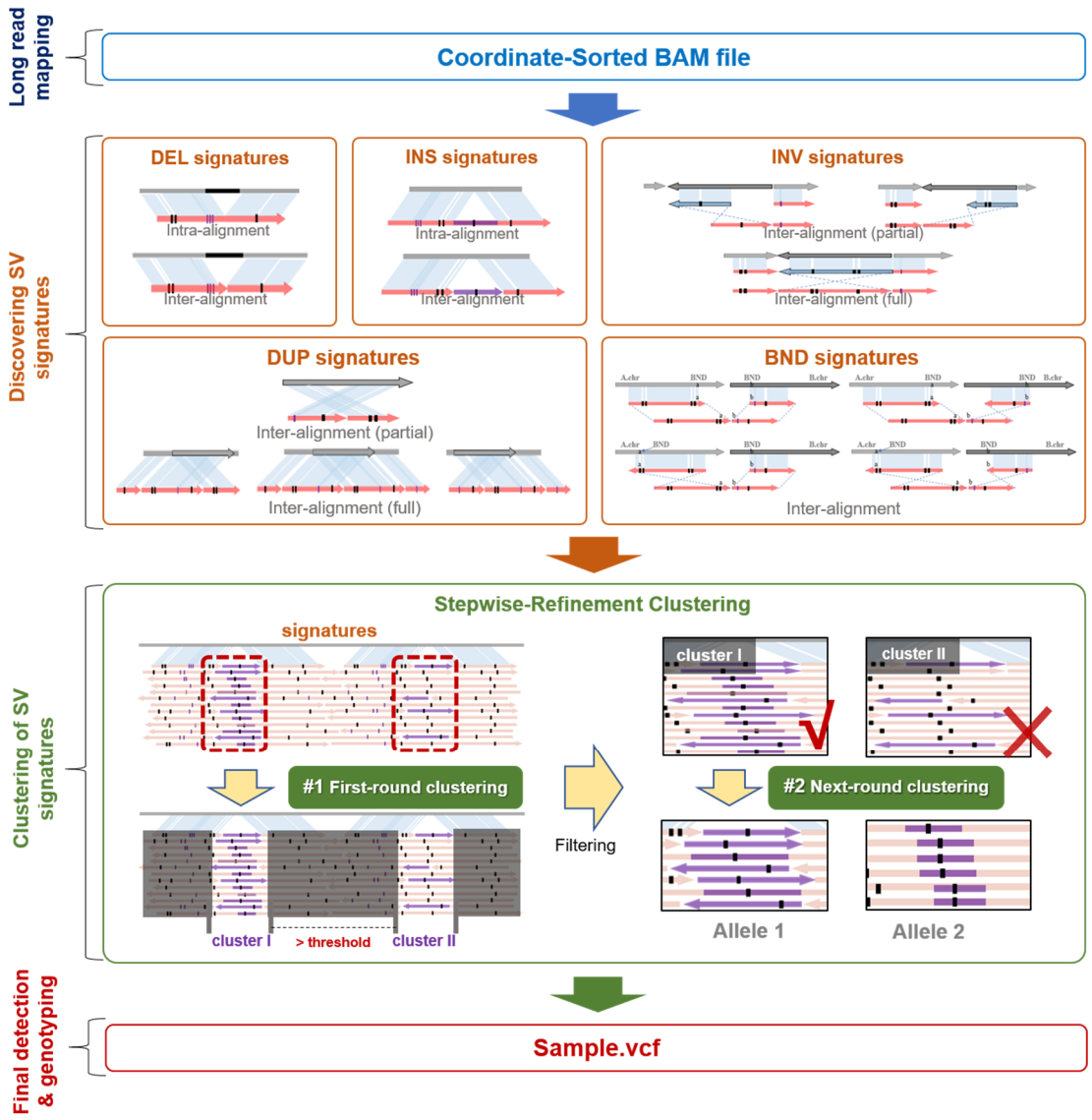

Figure 1 Scheme of cuteSV workflow.

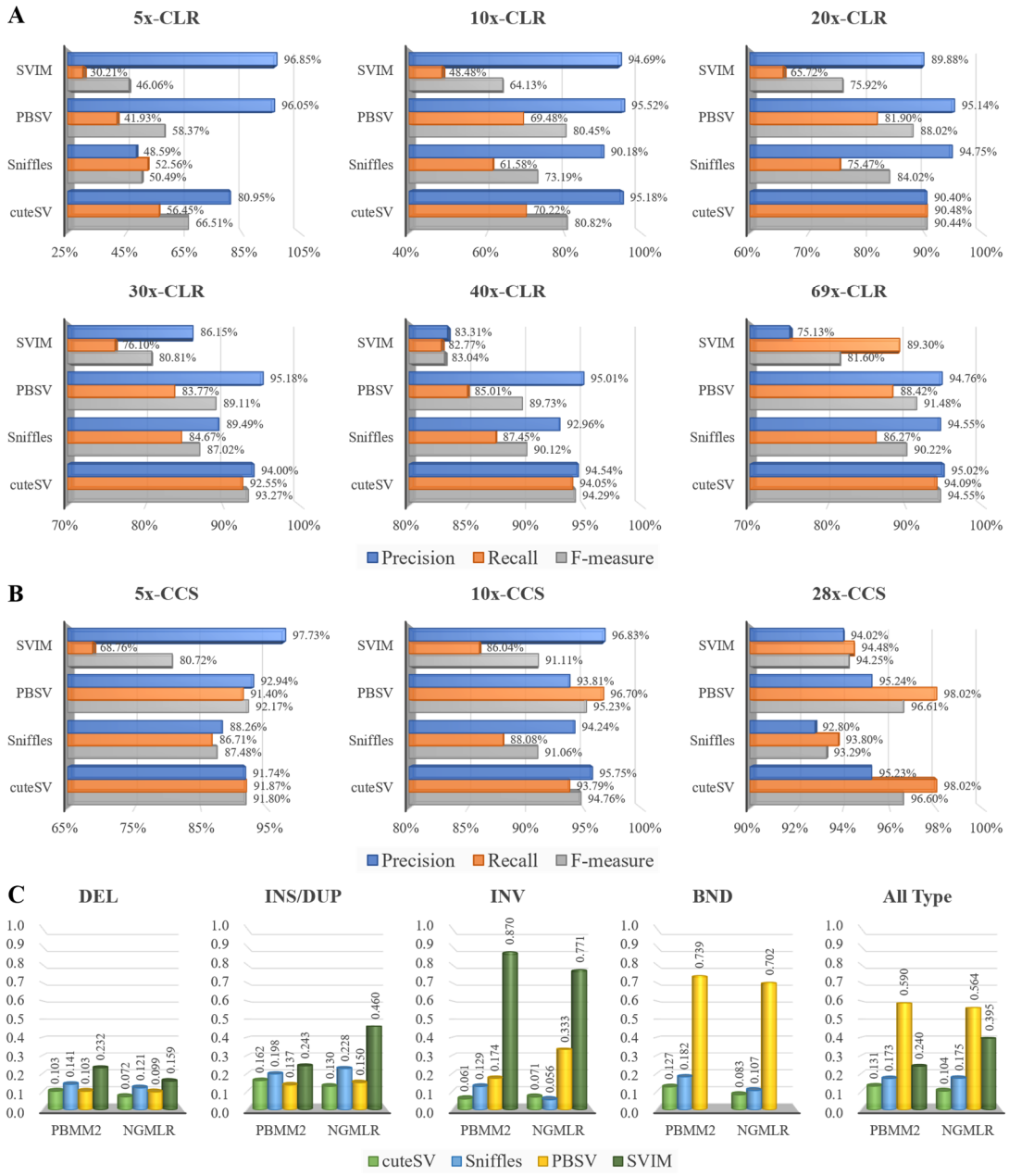

Figure 2 Performance comparison with the HG002 PacBio data for long read-based SV callers. (**A**) Precision, recall, and F-measure on the HG002 PacBio CLR datasets. (**B**) Precision, recall, and F-measure on the HG002 PacBio CCS datasets. (**C**) Benchmark results of Mendelian-Discordance-Rate (MDR) on the GIAB Ashkenazi Trio PacBio CLR datasets under different SV classes and long-read mappers. The lower of the MDR is, the better Mendelian consistency is, and the SV detections are more accurate.

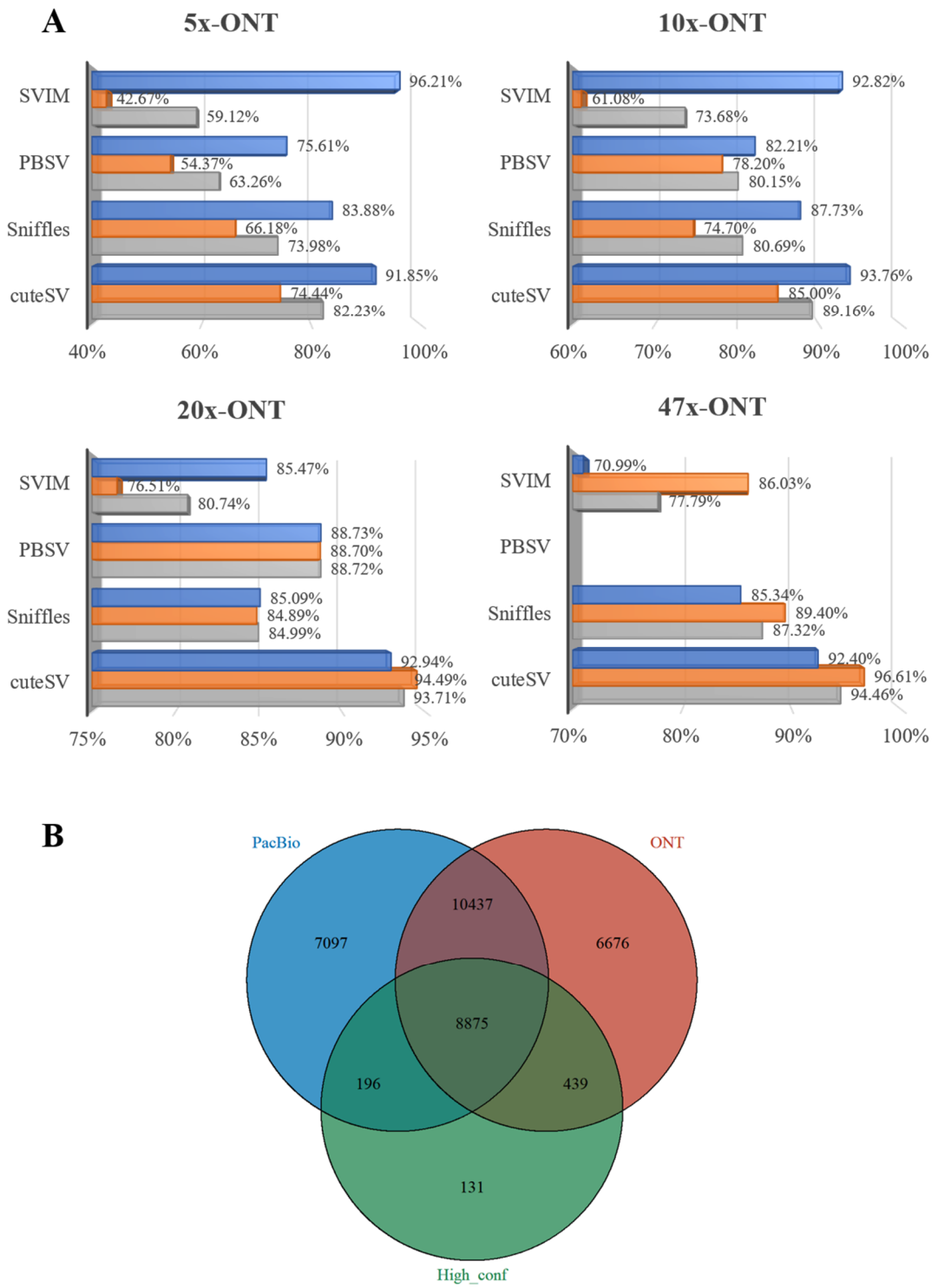

Figure 3 Performance comparison with the HG002 ONT PromethION data for long read-based SV callers. (**A**) Precision, recall, and F-measure on the HG002 ONT PromethION datasets. (**B**) The Venn diagram of SV calls in cuteSV on the HG002 PacBio CLR and ONT PromethION datasets, and the high confidence callsets. High_conf means the SV ground truth set of the HG002 human sample from GIAB.

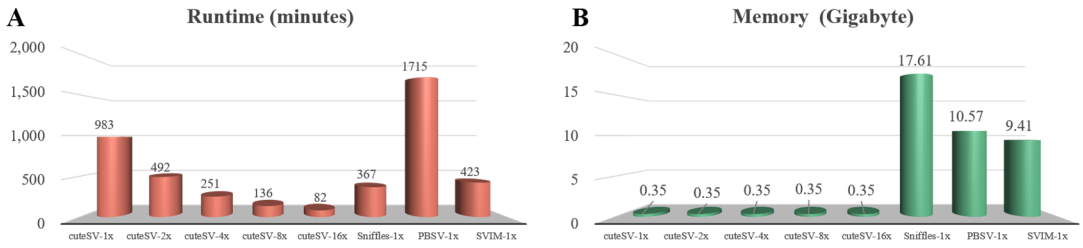

Figure 4 Computational performance comparison of long read-based SV callers. (**A**) The runtime under specific running thread. (**E**) The memory usage.

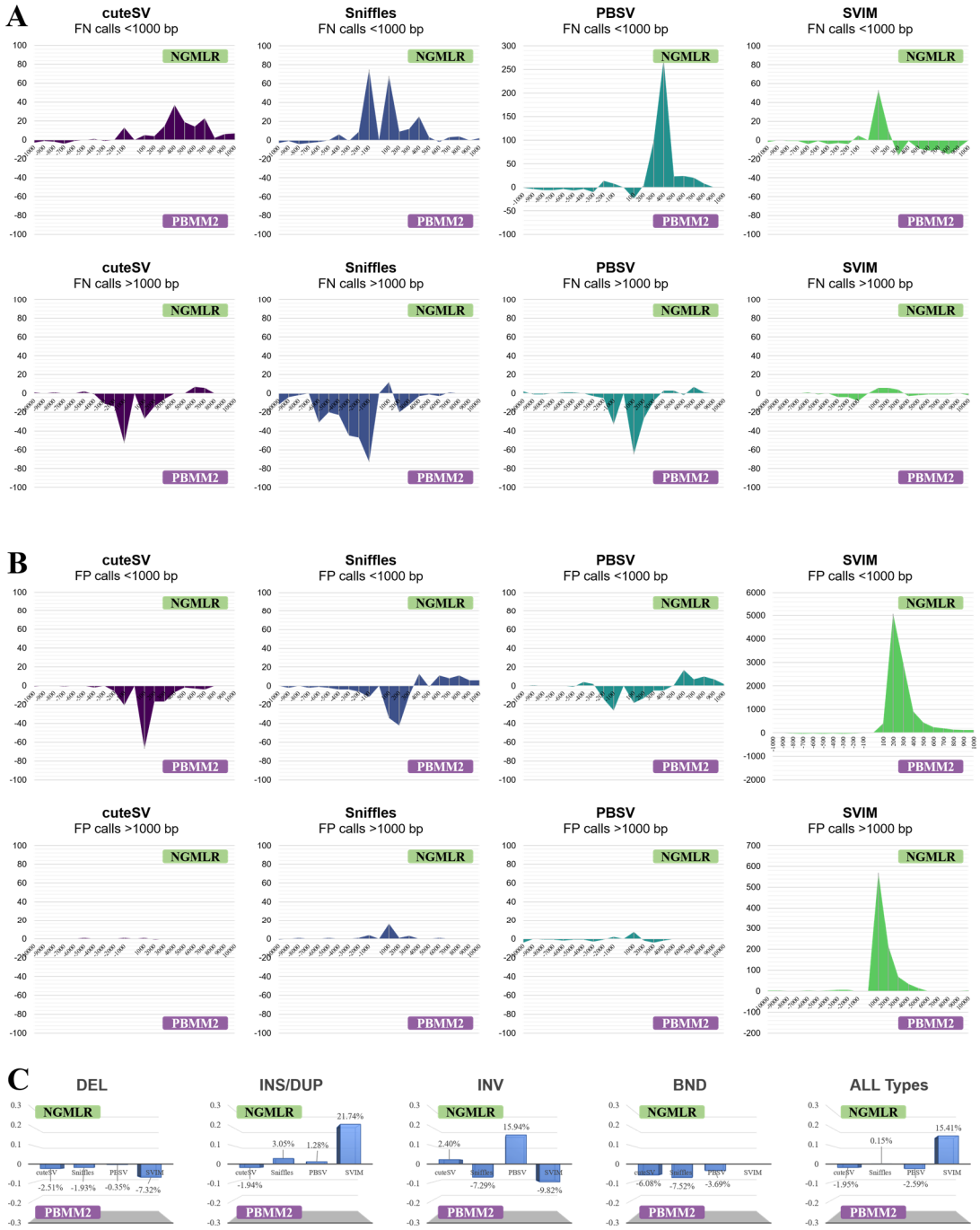

Figure 5 Performance comparison with different long-read aligners on the PacBio CLR dataset. (**A**) The amount of excess false-negative calls under different SV sizes with NGMLR and PBMM2. (**B**) The amount of excess false-positive calls under different SV sizes with NGMLR and PBMM2. (**C**) The different MDRs between NGMLR and PBMM2.

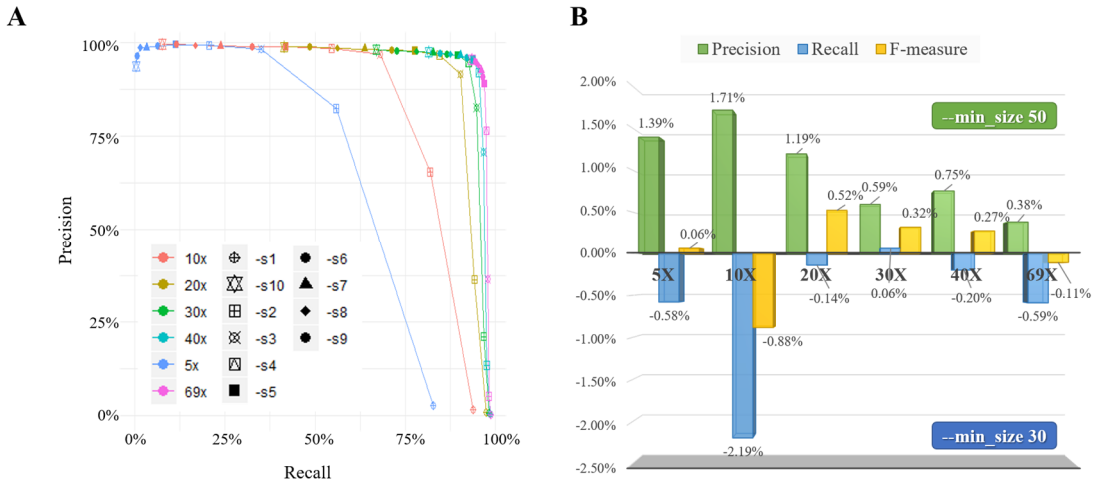

Figure 6 Comparison on the different parameters of *-s* (*--min_support*) and *–min_size* at the level of coverage. (**A**)The precision and recall under different support-read and sequencing coverages. (**B**) The difference of precision, recall, and F-measure in the best case of each coverage under *–min_size* 30 and 50.

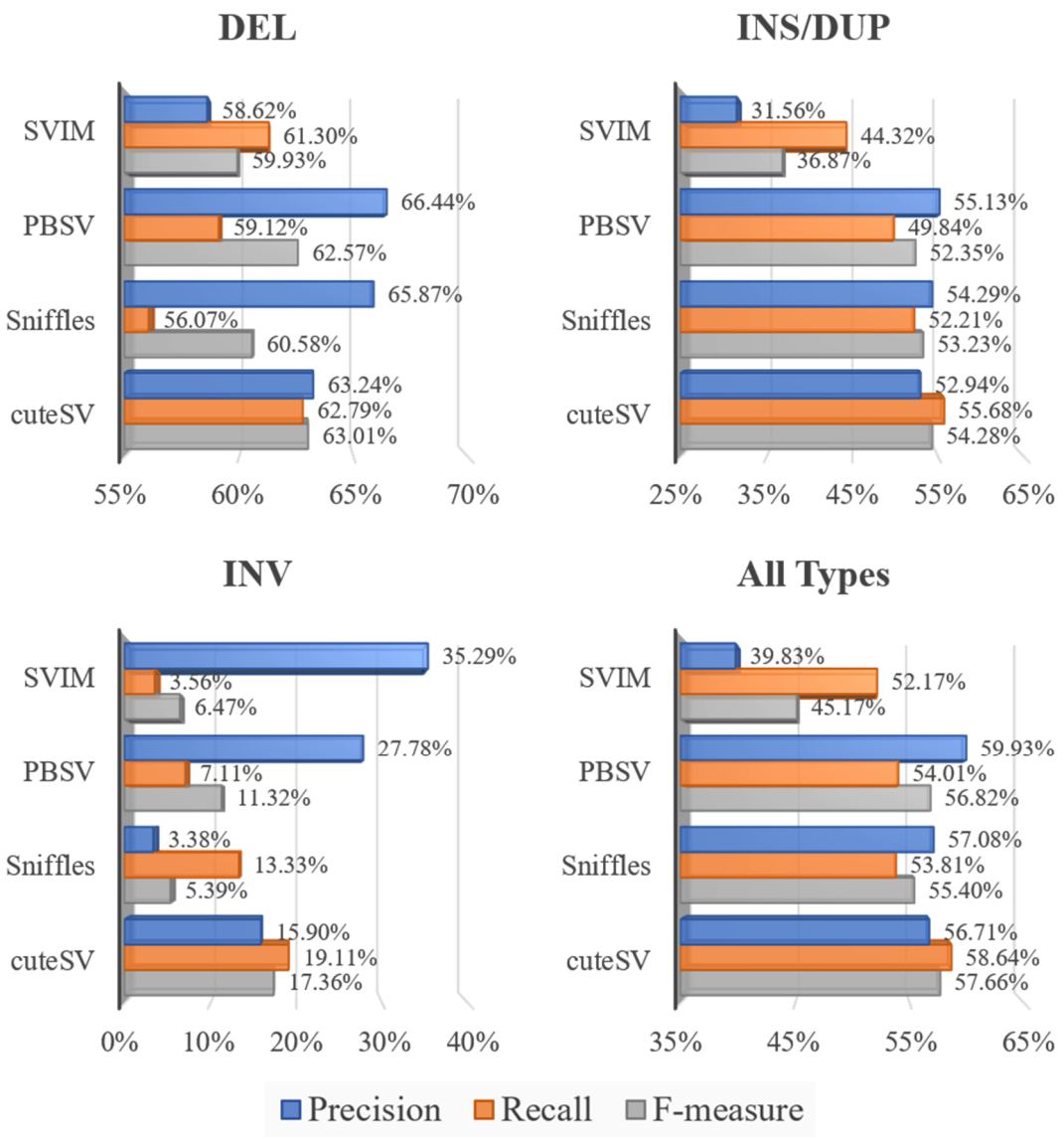

Figure 7 Performance comparison for the NA19240 human samples. Precision, recall, and F-measure on the NA19240 PacBio CLR datasets.